Automatic spelling correction for Russian social media texts

Alexey Sorokin[1,2,3], Tatiana Shavrina[1,3]

[1]Lomonosov Moscow State University, [2]Moscow Institute of Science and Technology, [3]General Internet Corpus of Russian

Dialogue
22nd International Conference on Computational Linguistics
Moscow, RSUH, 4th June, 2016

Applications of spellchecking

- Spellchecking has broad applications:
  - Search queries correction, information extraction.
  - Text editors.
  - Preprocessing in NLP tasks (parsing, fact extraction, etc.)

Applications of spellchecking

- Spellchecking has broad applications:
  - Search queries correction, information extraction.
  - Text editors.
  - Preprocessing in NLP tasks (parsing, fact extraction, etc.)
- For social media (blogs, social networks, information sites) we also need text normalization (correction of informal expressions to their formal variants).
- Especially important — when creating annotated corpora of social media.

Applications of spellchecking

- Spellchecking has broad applications:
  - Search queries correction, information extraction.
  - Text editors.
  - Preprocessing in NLP tasks (parsing, fact extraction, etc.)
- For social media (blogs, social networks, information sites) we also need text normalization (correction of informal expressions to their formal variants).
- Especially important — when creating annotated corpora of social media.
  - Parsing quality drastically falls without normalization.
  - The distribution of normalized/unnormalized variants is also important

Problems with spellchecking for social media

- High percentage of spontaneous errors and low quality of orthography.

Alexey Sorokin, Tatiana Shavrina
[1]Lomonosov Moscow State University, [2]Moscow Institute of Science and Technology, [3]General Internet Corpus of Russia

Problems with spellchecking for social media

- High percentage of spontaneous errors and low quality of orthography.
- Plenty of personal names, often colloquial.
- Lots of informal expressions.

Мы с Жекой решили поселицца в Зелике

Alexey Sorokin, Tatiana Shavrina
[1]Lomonosov Moscow State University, [2]Moscow Institute of Science and Technology, [3]General Internet Corpus of Russia

Problems with spellchecking for social media

- High percentage of spontaneous errors and low quality of orthography.
- Plenty of personal names, often colloquial.
- Lots of informal expressions.

Мы с Жекой решили поселицца в Зелике

- Orthography variation for expressing emotions:

Ну ты ваще крассотка, прям ооочень!

Alexey Sorokin, Tatiana Shavrina
[1]Lomonosov Moscow State University, [2]Moscow Institute of Science and Technology, [3]General Internet Corpus of Russia

Standard scheme of error correction

- First step: generate a list of candidates for every potential typo.
- Candidates — all dictionary words obtained by one edit.

Standard scheme of error correction

- First step: generate a list of candidates for every potential typo.
- Candidates — all dictionary words obtained by one edit.
- Problem one: there could be several candidates:

$$\textit{сврой} \mapsto \textit{сворой}, \textit{свой}, \textit{своей}, \textit{строй}, \textit{сварой}, \ldots$$

Alexey Sorokin, Tatiana Shavrina
[1]Lomonosov Moscow State University, [2]Moscow Institute of Science and Technology, [3]General Internet Corpus of Russia

Standard scheme of error correction

- First step: generate a list of candidates for every potential typo.
- Candidates — all dictionary words obtained by one edit.
- Problem one: there could be several candidates:

$$\textit{сврой} \mapsto \textit{своро́й},\ \textit{свой},\ \textit{своей},\ \textit{строй},\ \textit{сварой},\ \ldots$$

- Problem two: correct word is on large Levenstein distance

$$\begin{aligned} \textit{ваще} &\mapsto \textit{вообще}, \\ \textit{тока} &\mapsto \textit{только}, \\ \textit{поиграцца} &\mapsto \textit{поигра́ться}, \\ \textit{сиребренный} &\mapsto \textit{серебряный} \end{aligned}$$

Alexey Sorokin, Tatiana Shavrina
[1]Lomonosov Moscow State University, [2]Moscow Institute of Science and Technology, [3]General Internet Corpus of Russia

Standard scheme of error correction

- First step: generate a list of candidates for every potential typo.
- Candidates — all dictionary words obtained by one edit.
- Problem one: there could be several candidates:

$$\textit{сврой} \mapsto \textit{сворой, свой, своей, строй, сварой, } \ldots$$

- Problem two: correct word is on large Levenstein distance

$$\begin{aligned} \textit{ваще} &\mapsto \textit{вообще}, \\ \textit{тока} &\mapsto \textit{только}, \\ \textit{поиграцца} &\mapsto \textit{поиграться}, \\ \textit{сиребренный} &\mapsto \textit{серебряный} \end{aligned}$$

- Solution: measure not only graphic, but also phonetic similarity.

Alexey Sorokin, Tatiana Shavrina
[1]Lomonosov Moscow State University, [2]Moscow Institute of Science and Technology, [3]General Internet Corpus of Russia

Standard scheme of error correction

- First step: generate a list of candidates for every potential typo.
- Candidates — all dictionary words obtained by one edit.
- Problem one: there could be several candidates:

$$\textit{сврой} \mapsto \textit{сворой, свой, своей, строй, сварой, \ldots}$$

- Problem two: correct word is on large Levenstein distance

$$\begin{aligned} \textit{ваще} &\mapsto \textit{вообще}, \\ \textit{тока} &\mapsto \textit{только}, \\ \textit{поиграцца} &\mapsto \textit{поиграться}, \\ \textit{сиребренный} &\mapsto \textit{серебряный} \end{aligned}$$

- Solution: measure not only graphic, but also phonetic similarity.
- Dictionary words also contain typos

$$\begin{aligned} \textit{умолять} &\leftrightarrow \textit{умалять}, \\ \textit{кампания} &\leftrightarrow \textit{компания}, \\ \textit{сваей} &\mapsto \textit{стаей, своей, свай} \end{aligned}$$

Alexey Sorokin, Tatiana Shavrina
[1]Lomonosov Moscow State University, [2]Moscow Institute of Science and Technology, [3]General Internet Corpus of Russia

Candidate generation

We generate candidates for *every* word in the sentence:

- Words on Levenstein distance 1 (approximate search in prefix trie).

Candidate generation

We generate candidates for *every* word in the sentence:

- Words on Levenstein distance 1 (approximate search in prefix trie).
- Words having the same phonetic code (Metaphone-like algorithm):

Candidate generation

We generate candidates for *every* word in the sentence:

- Words on Levenstein distance 1 (approximate search in prefix trie).
- Words having the same phonetic code (Metaphone-like algorithm):
  Encoding table:

1	а,о,ы,у,я	3	и,е,ё,ю,я,э
5	б,п	6	в,ф
7	д,т	8	г,к,х
9	л	10	р
11	м	12	н
13	з,с	14	й
15	щ,ч	16	ж,ш
17	ц		

Candidate generation

We generate candidates for *every* word in the sentence:

- Words on Levenstein distance 1 (approximate search in prefix trie).
- Words having the same phonetic code (Metaphone-like algorithm):
  Encoding table:

1	а,о,ы,у,я	3	и,е,ё,ю,я,э
5	б,п	6	в,ф
7	д,т	8	г,к,х
9	л	10	р
11	м	12	н
13	з,с	14	й
15	щ,ч	16	ж,ш
17	ц		

  other phonetic transformations:
  - ь, ъ omitted, affecting only next vowel.

Alexey Sorokin, Tatiana Shavrina
[1]Lomonosov Moscow State University, [2]Moscow Institute of Science and Technology, [3]General Internet Corpus of Russia

Candidate generation

We generate candidates for *every* word in the sentence:

- Words on Levenstein distance 1 (approximate search in prefix trie).
- Words having the same phonetic code (Metaphone-like algorithm):
  Encoding table:

1	а,о,ы,у,я	3	и,е,ё,ю,я,э
5	б,п	6	в,ф
7	д,т	8	г,к,х
9	л	10	р
11	м	12	н
13	з,с	14	й
15	щ,ч	16	ж,ш
17	ц		

  other phonetic transformations:
  - *ь*, *ъ* omitted, affecting only next vowel.
  - *тьс*, *тс* $\mapsto$ *ц*, *т* omitted between two consonants.

Alexey Sorokin, Tatiana Shavrina
[1]Lomonosov Moscow State University, [2]Moscow Institute of Science and Technology, [3]General Internet Corpus of Russia

Sentence generating

We generate candidates for *every* word in the sentence:

Стадо	свией	бросилось	со	скалы	в	море
стадо	свиней своей сваей	бросилось	со сто до то	скалы шкалы скулы	в во с	море мире горе торе

Alexey Sorokin, Tatiana Shavrina
[1]Lomonosov Moscow State University, [2]Moscow Institute of Science and Technology, [3]General Internet Corpus of Russia

Sentence generating

We generate candidates for *every* word in the sentence:

Стадо	свией	бросилось	со	скалы	в	море
стадо	свиней своей сваей	бросилось	со сто до то	скалы шкалы скулы	в во с	море мире горе торе

- Possible sentences:

  стадо свиней бросилось со скалы в море

  стадо сваей бросилось со скалы в море

  ...

  стадо своей бросилось со скулы в хоре

Alexey Sorokin, Tatiana Shavrina

[1]Lomonosov Moscow State University, [2]Moscow Institute of Science and Technology, [3]General Internet Corpus of Russia

Sentence generating

We generate candidates for *every* word in the sentence:

Стадо	свией	бросилось	со	скалы	в	море
стадо	свиней своей сваей	бросилось	со сто до то	скалы шкалы скулы	в во с	море мире горе торе

- Possible sentences:

  стадо свиней бросилось со скалы в море

  стадо сваей бросилось со скалы в море

  ...

  стадо своей бросилось со скулы в хоре

- There are too many sentences, we cannot rank all.

Alexey Sorokin, Tatiana Shavrina

[1]Lomonosov Moscow State University, [2]Moscow Institute of Science and Technology, [3]General Internet Corpus of Russia

Sentence generating

We generate candidates for *every* word in the sentence:

Стадо	свией	бросилось	со	скалы	в	море
стадо	свиней своей сваей	бросилось	со сто до то	скалы шкалы скулы	в во с	море мире горе торе

- Possible sentences:

  стадо свиней бросилось со скалы в море

  стадо сваей бросилось со скалы в море

  ...

  стадо своей бросилось со скулы в хоре

- There are too many sentences, we cannot rank all.
- By the way, how we rank them?

Alexey Sorokin, Tatiana Shavrina
[1]Lomonosov Moscow State University, [2]Moscow Institute of Science and Technology, [3]General Internet Corpus of Russia

Baseline ranking scheme

- Formula for best sentence selection:

$$\widehat{t} = \text{argmax}_t\, p(t|s) = \text{argmax}_t\, p(s|t)p(t)$$

Baseline ranking scheme

- Formula for best sentence selection:

$$\widehat{t} = \operatorname{argmax}_t p(t|s) = \operatorname{argmax}_t p(s|t)p(t)$$

$p(s|t)$ — edit model probability to get s from t,

$p(t)$ — language model probability.

Alexey Sorokin, Tatiana Shavrina
[1]Lomonosov Moscow State University, [2]Moscow Institute of Science and Technology, [3]General Internet Corpus of Russia

Baseline ranking scheme

- Formula for best sentence selection:

$$\widehat{t} = \text{argmax}_t\, p(t|s) = \text{argmax}_t\, p(s|t)p(t)$$

$p(s|t)$ — edit model probability to get s from t,

$p(t)$ — language model probability.

- Both probabilities are multiplicative:

$$p(s|t) = \prod_i p(s_{(i)}|t_{(i)})$$

$$p(t) = p(t_1)p(t_2|t_1)p(t_3|t_1\,t_2)\ldots p(t_n|t_{n-2}\,t_{n-1})$$

Alexey Sorokin, Tatiana Shavrina
[1]Lomonosov Moscow State University, [2]Moscow Institute of Science and Technology, [3]General Internet Corpus of Russia

Baseline ranking scheme

- Formula for best sentence selection:

$$\widehat{t} = \operatorname{argmax}_t p(t|s) = \operatorname{argmax}_t p(s|t)p(t)$$

$p(s|t)$ — edit model probability to get s from t,

$p(t)$ — language model probability.

- Both probabilities are multiplicative:

$$p(s|t) = \prod_i p(s_{(i)}|t_{(i)})$$

$$p(t) = p(t_1)p(t_2|t_1)p(t_3|t_1\,t_2)\ldots p(t_n|t_{n-2}\,t_{n-1})$$

- Log-probabilities:

$$\widehat{t} = \log p(s|t) + \log p(t)$$

$$\log p(s|t) = \sum_i \log p(s_{(i)}|t_{(i)})$$

$$\log p(t) = \log p(t_1) + \log p(t_2|t_1) + \ldots + \log p(t_n|t_{n-2}\,t_{n-1})$$

Baseline ranking scheme

- Formula for best sentence selection:

$$\widehat{t} = \operatorname{argmax}_t p(t|s) = \operatorname{argmax}_t p(s|t)p(t)$$

$p(s|t)$ — edit model probability to get s from t,

$p(t)$ — language model probability.

- Both probabilities are multiplicative:

$$p(s|t) = \prod_i p(s_{(i)}|t_{(i)})$$

$$p(t) = p(t_1)p(t_2|t_1)p(t_3|t_1\,t_2)\ldots p(t_n|t_{n-2}\,t_{n-1})$$

- Log-probabilities:

$$\widehat{t} = \log p(s|t) + \log p(t)$$

$$\log p(s|t) = \sum_i \log p(s_{(i)}|t_{(i)})$$

$$\log p(t) = \log p(t_1) + \log p(t_2|t_1) + \ldots + \log p(t_n|t_{n-2}\,t_{n-1})$$

- Scores are additive, A^*-search could be used.

Baseline ranking scheme

- We used A^*-search for hypotheses selection.

Baseline ranking scheme

- We used A^*-search for hypotheses selection.
- We update hypotheses list after each word.

Baseline ranking scheme

- We used A^*-search for hypotheses selection.
- We update hypotheses list after each word.
- Update procedure:
  - Find all possible ways to replace last words by one or two dictionary words or keep it untouched.

Alexey Sorokin, Tatiana Shavrina
[1]Lomonosov Moscow State University, [2]Moscow Institute of Science and Technology, [3]General Internet Corpus of Russia

Baseline ranking scheme

- We used A^*-search for hypotheses selection.
- We update hypotheses list after each word.
- Update procedure:
  - Find all possible ways to replace last words by one or two dictionary words or keep it untouched.
  - Find all possible ways to replace last 2 words by a single one.

Baseline ranking scheme

- We used A^*-search for hypotheses selection.
- We update hypotheses list after each word.
- Update procedure:
  - Find all possible ways to replace last words by one or two dictionary words or keep it untouched.
  - Find all possible ways to replace last 2 words by a single one.
  - Update hypotheses from last 2 steps by attaching current candidates.

Baseline ranking scheme

- We used A^*-search for hypotheses selection.
- We update hypotheses list after each word.
- Update procedure:
  - Find all possible ways to replace last words by one or two dictionary words or keep it untouched.
  - Find all possible ways to replace last 2 words by a single one.
  - Update hypotheses from last 2 steps by attaching current candidates.
- Keep 50 best hypotheses after each step.

Baseline ranking scheme

- We used A^*-search for hypotheses selection.
- We update hypotheses list after each word.
- Update procedure:
  - Find all possible ways to replace last words by one or two dictionary words or keep it untouched.
  - Find all possible ways to replace last 2 words by a single one.
  - Update hypotheses from last 2 steps by attaching current candidates.
- Keep 50 best hypotheses after each step.
- Baseline edit model:
  - At most one change by Leveshtein model or the same phonetic code.

Baseline ranking scheme

- We used A^*-search for hypotheses selection.
- We update hypotheses list after each word.
- Update procedure:
  - Find all possible ways to replace last words by one or two dictionary words or keep it untouched.
  - Find all possible ways to replace last 2 words by a single one.
  - Update hypotheses from last 2 steps by attaching current candidates.
- Keep 50 best hypotheses after each step.
- Baseline edit model:
  - At most one change by Leveshtein model or the same phonetic code.
  - Euristically defined probabilities of different changes.

Reranking scheme

- We used baseline model only to select candidates.

Reranking scheme

- We used baseline model only to select candidates.
- We rerank these hypotheses with features.

Reranking scheme

- We used baseline model only to select candidates.
- We rerank these hypotheses with features.
- Reranking procedure:
  - For the sentences in the training set generate all hypothesis and calculate their feature scores.

Reranking scheme

- We used baseline model only to select candidates.
- We rerank these hypotheses with features.
- Reranking procedure:
  - For the sentences in the training set generate all hypothesis and calculate their feature scores.
  - Let $\mathbf{x}$ be the feature vector of the correct hypothesis and $\mathbf{x_1}, \mathbf{x_r}$ of the others.

Reranking scheme

- We used baseline model only to select candidates.
- We rerank these hypotheses with features.
- Reranking procedure:
  - For the sentences in the training set generate all hypothesis and calculate their feature scores.
  - Let $\mathbf{x}$ be the feature vector of the correct hypothesis and $\mathbf{x_1}, \mathbf{x_r}$ of the others.

$$\mathbf{x} - \mathbf{x_1}, \ldots, \mathbf{x} - \mathbf{x_r} \quad \text{– positive instances,}$$
$$\mathbf{x_1} - \mathbf{x}, \ldots, \mathbf{x_r} - \mathbf{x} \quad \text{– negative instances}$$

Reranking scheme

- We used baseline model only to select candidates.
- We rerank these hypotheses with features.
- Reranking procedure:
  - For the sentences in the training set generate all hypothesis and calculate their feature scores.
  - Let $\mathbf{x}$ be the feature vector of the correct hypothesis and $\mathbf{x_1}, \mathbf{x_r}$ of the others.

    $\mathbf{x} - \mathbf{x_1}, \ldots, \mathbf{x} - \mathbf{x_r}$ – positive instances,

    $\mathbf{x_1} - \mathbf{x}, \ldots, \mathbf{x_r} - \mathbf{x}$ – negative instances
  - Logistic regression is trained on these instances.

Alexey Sorokin, Tatiana Shavrina
[1]Lomonosov Moscow State University, [2]Moscow Institute of Science and Technology, [3]General Internet Corpus of Russia

Reranking scheme

- We used baseline model only to select candidates.
- We rerank these hypotheses with features.
- Reranking procedure:
  - For the sentences in the training set generate all hypothesis and calculate their feature scores.
  - Let $\mathbf{x}$ be the feature vector of the correct hypothesis and $\mathbf{x_1}, \mathbf{x_r}$ of the others.

    $\mathbf{x} - \mathbf{x_1}, \ldots, \mathbf{x} - \mathbf{x_r}$ – positive instances,

    $\mathbf{x_1} - \mathbf{x}, \ldots, \mathbf{x_r} - \mathbf{x}$ – negative instances
  - Logistic regression is trained on these instances.
- In testing phase we select the sentence with highest score from the reranker.

Alexey Sorokin, Tatiana Shavrina
[1]Lomonosov Moscow State University, [2]Moscow Institute of Science and Technology, [3]General Internet Corpus of Russia

Features

	Description		Description
F1	Sentence length in words	F2	Edit model score
F3	Language model score	F4	N_{corr} (total)
F5	Number of OOV words	F6	N_{corr} (OOV words)
F7	N_{corr} (dictionary words)	F8	N_{corr} (capitalized words)
F9	N_{corr} (edit distance 1)	F10	N_{corr} (phonetic similarity)
F11	N_{corr} (word lists)	F12	N_{corr} (space insertions)
F12	N_{corr} (space deletions)	F14	OOV words with dictionary partitions
F15	Morpho model score	F16	Weighted edit distance score
F17	Semantic model score	F18	Semantic model score
F19	Prep. model score	F20	Prep. model score

Features

	Description		Description
F1	Sentence length in words	F2	Edit model score
F3	Language model score	F4	N_{corr} (total)
F5	Number of OOV words	F6	N_{corr} (OOV words)
F7	N_{corr} (dictionary words)	F8	N_{corr} (capitalized words)
F9	N_{corr} (edit distance 1)	F10	N_{corr} (phonetic similarity)
F11	N_{corr} (word lists)	F12	N_{corr} (space insertions)
F12	N_{corr} (space deletions)	F14	OOV words with dictionary partitions
F15	Morpho model score	F16	Weighted edit distance score
F17	Semantic model score	F18	Semantic model score
F19	Prep. model score	F20	Prep. model score

F15, F17-F20 not used in final testing

Feature and data description

Features:

- Weighted Levenshtein distance: Brill-Moore model on edit frequencies.

Feature and data description

Features:

- Weighted Levenshtein distance: Brill-Moore model on edit frequencies.
- Morpho-model: trigram models on morpho tags (with case, number and gender for nouns and adjectives).

Feature and data description

Features:

- Weighted Levenshtein distance: Brill-Moore model on edit frequencies.
- Morpho-model: trigram models on morpho tags (with case, number and gender for nouns and adjectives).
- Semantic model: cooccurrence probability on word pairs (Schaback, 2009).

Feature and data description

Features:

- Weighted Levenshtein distance: Brill-Moore model on edit frequencies.
- Morpho-model: trigram models on morpho tags (with case, number and gender for nouns and adjectives).
- Semantic model: cooccurrence probability on word pairs (Schaback, 2009).
- Prepositional model: number of nouns in case appropriate for preposition.

Feature and data description

Features:

- Weighted Levenshtein distance: Brill-Moore model on edit frequencies.
- Morpho-model: trigram models on morpho tags (with case, number and gender for nouns and adjectives).
- Semantic model: cooccurrence probability on word pairs (Schaback, 2009).
- Prepositional model: number of nouns in case appropriate for preposition.

Training data:

- Weighted Levenstein distance: training data from organizers.

Feature and data description

Features:

- Weighted Levenshtein distance: Brill-Moore model on edit frequencies.
- Morpho-model: trigram models on morpho tags (with case, number and gender for nouns and adjectives).
- Semantic model: cooccurrence probability on word pairs (Schaback, 2009).
- Prepositional model: number of nouns in case appropriate for preposition.

Training data:

- Weighted Levenstein distance: training data from organizers.
- Language model: trigrams on 25000000 word LJ corpus.
- Morpho-model: tag trigrams on 25000000 word LJ corpus.

Feature and data description

Features:

- Weighted Levenshtein distance: Brill-Moore model on edit frequencies.
- Morpho-model: trigram models on morpho tags (with case, number and gender for nouns and adjectives).
- Semantic model: cooccurrence probability on word pairs (Schaback, 2009).
- Prepositional model: number of nouns in case appropriate for preposition.

Training data:

- Weighted Levenstein distance: training data from organizers.
- Language model: trigrams on 25000000 word LJ corpus.
- Morpho-model: tag trigrams on 25000000 word LJ corpus.
- Semantic model — the same training corpus.

Alexey Sorokin, Tatiana Shavrina
[1]Lomonosov Moscow State University, [2]Moscow Institute of Science and Technology, [3]General Internet Corpus of Russia

Results on train data

Development set:

Model	Precision	Recall	F1-measure	Sentence accuracy
Baseline	71.68	78.01	74.71	70.70
Weighted baseline	82.83	77.89	80.28	79.40
Levenshtein	87.69	79.15	83.20	81.90
Competition	**88.43**	81.44	84.79	**83.20**
Competition+Morpho	88.15	**81.79**	**84.85**	83.00

Results on test data

Development set:

Model	Precision	Recall	F1-measure	Sentence accuracy
Baseline	63.11	67.26	65.12	60.06
Weighted baseline	75.55	64.27	69.46	66.93
Levenshtein	80.58	65.94	72.53	68.63
Competition	**81.98**	**69.25**	**75.07**	**70.32**
Competition+Morpho	81.12	68.98	74.56	70.22
Second place	74.87	62.31	64.82	61.35

Uncorrected errors

- paronyms

  советую купить от кашля мукалтин а горло поласкать гидроперитом
  советую купить от кашля мукалтин а горло поласкать гидроперитом

  он не стал чьимто рабом и не перед кем так и не приклонился
  он не стал чьим-то рабом и не перед кем так и не приклонился

Alexey Sorokin, Tatiana Shavrina
[1]Lomonosov Moscow State University, [2]Moscow Institute of Science and Technology, [3]General Internet Corpus of Russia

Uncorrected errors

- paronyms

  советую купить от кашля мукалтин а горло поласкать гидроперитом
  советую купить от кашля мукалтин а горло поласкать гидроперитом

  он не стал чьимто рабом и не перед кем так и не приклонился
  он не стал чьим-то рабом и не перед кем так и не приклонился
- space insertions

  страшно представить еслиб с ней что-то случилось
  страшно представить если с ней что-то случилось

Alexey Sorokin, Tatiana Shavrina
[1]Lomonosov Moscow State University, [2]Moscow Institute of Science and Technology, [3]General Internet Corpus of Russia

Uncorrected errors

- paronyms
  - *советую купить от кашля мукалтин а горло поласкать гидроперитом*
  - *советую купить от кашля мукалтин а горло поласкать гидроперитом*
  - *он не стал чьимто рабом и не перед кем так и не приклонился*
  - *он не стал чьим-то рабом и не перед кем так и не приклонился*
- space insertions
  - *страшно представить еслиб с ней что-то случилось*
  - *страшно представить если с ней что-то случилось*
- wrong candidate selection
  - *довлю на слове правда есть подозрение что моя сессия может затянутся*
  - *давлю на слове правда есть подозрение что моя сессия может затянуться*

Alexey Sorokin, Tatiana Shavrina
[1]Lomonosov Moscow State University, [2]Moscow Institute of Science and Technology, [3]General Internet Corpus of Russia

Conclusions

- Reranking approach is optimal for Russian language spelling correction.

Conclusions

- Reranking approach is optimal for Russian language spelling correction.
- Using phonetic model improves search quality.
- Using weighted Levenshtein distance improves results.

Conclusions

- Reranking approach is optimal for Russian language spelling correction.
- Using phonetic model improves search quality.
- Using weighted Levenshtein distance improves results.
- Using simple morphology and semantics does not help.

Conclusions

- Reranking approach is optimal for Russian language spelling correction.
- Using phonetic model improves search quality.
- Using weighted Levenshtein distance improves results.
- Using simple morphology and semantics does not help.
- Therefore we need more complex model for these phenomena.